REMARKS

Status of Claims:

Claims 1-21 are present for examination.

Response to Arguments Section:

In the "Response to Arguments" section of the present Office Action, the Examiner stated that, "[i]n response to applicant's argument that the reference fails to show certain features of applica[nt's] invention, it is noted that [in] claims 1, 9, 20, and 21, the features upon which applicant relies (i.e. 'over a single socket connection ..') does not define in the specification." (Office Action; page 2).

Applicant submits that the feature "over a single socket connection" is defined in the specification. In particular, applicant points to page 4, line 30 to page 5, line 1 of the specification, where applicant states in connection with FIG. 1 that, "[t]he system 100 illustrates only one socket connection between a single client 102 and a single host 120." Applicant further explains about "a socket connection" on page 5, lines 20-28 of the specification. In addition, applicant states on page 6, lines 11-13, that, "[f]igure 2 shows a flow chart of a method 200 for maintaining two-way asynchronous communications over a socket connection between a client and a web server." At step 210 in FIG. 2, a socket connection is opened between the client and the web server, and at step 250 in FIG. 2, the socket connection is closed. (Specification; page 6, lines 14-15; page 7, lines 23-24).

In order to expedite prosecution of the present application, applicant is amending claims 1, 9, 20, and 21 to recite the feature, "over one socket connection", rather than, "over a single socket connection", because, "one socket connection", is clearly mentioned in the specification at page 4, line 30 to page 5, line 1. Applicant notes that "one socket connection" has the same meaning as "a single socket connection" and, thus, the amendments to the claims should be entered inasmuch as they require no new search or consideration.

Claim Rejections:

Claims 1-17 and 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rangarajan et al. (U.S. Patent No. 6,510,439) (hereinafter Rangarajan) in view of Cianfrocca et al. (U.S. Patent No. 6,088,796) (hereinafter Cianfrocca).

Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rangarajan in view of Cianfrocca, and further in view of Reisman (U.S. Patent No. 6,611,862).

With respect to claims 1-21, as amended, the rejections are respectfully traversed.

Independent claim 1, as amended, recites a method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction, comprising:

> "communicating an HTTP request from the client to the web server, wherein the HTTP request is configured to initialize a CGI that operates within or in conjunction with the web server; and
>
> executing operations associated with the CGI, wherein the operations are configured to perform the **two-way asynchronous** communication with the client over one socket connection until terminated by the client or the CGI." (Emphasis Added).

A method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction including the above-quoted features has at least the advantages that: (i) an **HTTP request** is communicated from the client to the web server, where the HTTP request is configured to initialize a CGI that operates within or in conjunction with the web server; and (ii) operations associated with the CGI are executed, where the operations are configured to perform the **two-way asynchronous** communication with the client over one socket connection until terminated by the client or the CGI. (Specification; page 3, lines 12-22; page 4, line 30 to page 5, line 1; page 6, line 11 to page 7, line 24; page 8, line 12 to page 9, line 12; FIGs. 1-4).

Neither Rangarajan nor Cianfrocca, alone or in combination, disclose or suggest a method including the above-quoted features for at least the following two reasons.

First, Rangarajan neither discloses nor suggests a method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction in which: (i) an HTTP request is communicated from the client to the web server, where the HTTP request is configured to initialize a CGI that operates within or in conjunction with the web server; and (ii) operations associated with the CGI are executed, where the operations are configured to perform the **two-way asynchronous** communication with the client.

The Examiner states that, "Rangarajan teaches a method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction". For such a teaching, the Examiner points to Rangarajan, column 2, lines 3-6, column 6, lines 55-67, and column 7, lines 10-13. Each of the cited portions of Rangarajan will now be examined in detail in order to make it clear that Rangarajan does not teach a method of maintaining **two-way asynchronous** communication between a client and a web server using a single HTTP transaction.

In column 2, lines 1-14, Rangarajan states the following:

> "One newly developed technique relies on the notion of group consistency within a persistent HTTP connection. Under this consistency model, when a client accesses a group of interrelated documents within a **single persistent HTTP connection**, it receives a consistent version of all documents in the group, even if some of the documents are updated during the access interval. Access to the correct version of a file is provided by selectively updating and reloading the file server's request Redirect data table. This technique is discussed in more detail in S. Rangarajan, S. Yajnik, and P. Jalote, '**WCP--A tool for consistent on-line update of documents in a WWW server**', Proceedings of the Conference on the World-Wide Web (WWW7), April 1998, Brisbane, Australia."
> (Rangarajan; column 2, lines 1-14) (Emphasis Added).

Applicant assumes that the Examiner believes that the single persistent HTTP connection recited in Rangarajan allows for two-way asynchronous communication. However, the **single**

persistent HTTP connection recited in Rangarajan does **not** allow for **two-way asynchronous** communication. (Rangarajan; column 2, lines 1-14). Rangarajan does not provide a definition for a single persistent HTTP connection in the Rangarajan patent reference, so it is necessary to look to the article (S. Rangarajan, S. Yajnik, and P. Jalote, "WCP--A tool for consistent on-line update of documents in a WWW server", Proceedings of the Conference on the World-Wide Web (WWW7), April 1998, Brisbane, Australia) cited in the Rangarajan patent reference for a definition of a single persistent HTTP connection.

Applicant has obtained what applicant believes to be an Internet downloadable version of the article cited in the Ragarajan patent reference, which is S. Rangarajan, S. Yajnik, and P. Jalote, "WCP--A tool for consistent on-line update of documents in a WWW server", Computer Networks and ISDN Systems, Vol. 30, Issues 1-7, Article: FP32, Proceedings of the Conference on the World-Wide Web (WWW7), April 1998 (hereinafter Rangarajan Web Article). A copy of the Ragarajan Web Article is provided in an Information Disclosure Statement submitted concurrently with this reply.

In the Rangarajan Web Article, a single persistent HTTP connection is simply a connection using the **HTTP/1.1 protocol**. (Rangarajan Web Article; page 2, lines 18-33; page 4, lines 32-44). It is important to understand that the **HTTP/1.1** protocol described in the Rangarajan Web Article does **not** allow for two-way **asynchronous** communication between a client and a server using a single HTTP transaction. (Rangarajan Web Article; page 4, lines 32-44). Instead, the HTTP/1.1 protocol described in the Rangarajan Web Article only allows for **synchronous** communication between a client and a server using a single HTTP transaction. (Rangarajan Web Article; page 4, lines 32-44).

Indeed, the Rangarajan Web Article defines the HTTP/1.1 protocol as a "**request/response** protocol." (Rangarajan Web Article; page 4, lines 33-35) (Emphasis Added). As such, in the HTTP/1.1 protocol, a client sends a **request** to a server, and then the server **responds** to the request of the client. (Rangarajan Web Article; page 4, lines 35-38). There is only a cause and effect relationship between a request and a response in the HTTP/1.1 protocol,

and the sending of information cannot be decoupled from the receiving of a request in the HTTP/1.1 protocol. (Rangarajan Web Article; page 4, lines 32-44). As a result, the HTTP/1.1 protocol only allows for **synchronous** communication using a single HTTP transaction.

A connection using the HTTP/1.1 protocol is persistent in that the connection remains open unless explicitly closed by either the server or the client. (Rangarajan Web Article; page 4, lines 41-42). However, when a persistent connection using the HTTP/1.1 protocol is open, only **synchronous** communication occurs over the connection, where the client sends a request and then the server responds to the request. (Rangarajan Web Article; page 4, lines 43-44). Such a persistent connection using the HTTP/1.1 protocol only allows for sequential request/reply operations and, as a consequence, a connection using the HTTP/1.1 protocol does not allow for **asynchronous** communication.

In contrast, a significant feature of embodiments of applicants' invention is the ability to maintain two-way **asynchronous** communication between a client and a web server using a single HTTP transaction. As shown in applicants' FIG. 2, **asynchronous** communication allows for operations in which the CGI reads and processes client requests (steps 230, 240), and such reading and processing is decoupled from operations in which the CGI determines if there is information or requests to send to the client and operations in which the CGI sends information or requests to the client (steps 232, 242). Also, as shown in applicants' FIG. 4, **asynchronous** communication allows for operations in which the client sends client requests or information (step 430), and such sending is decoupled from operations in which the client receives information or requests and operations in which the client processes received information or requests (steps 432, 437).

Thus, with two-way **asynchronous** communication, there does not have to be a cause and effect relationship in operations of the CGI between reading client requests and sending information to the client. Also, with two-way asynchronous communication, there does not have to be a cause and effect relationship in operations of the client between sending client requests and receiving information. For example, information may be sent from the CGI to the client

where the information is not sent directly in response to a client request, and operations for reading requests and sending information by the CGI may occur concurrently. (Applicants' FIGs. 2 and 4; Specification; page 3, lines 5-7 and 12-22; page 6, lines 5-24; page 8, line 27 to page 9, line 7).

In order to more fully explain the difference between **synchronous** and **asynchronous** communication, consider the example of a telephone conversation between two parties, which is a simplified example of two-way asynchronous communication. A telephone conversation is an example of two-way asynchronous communication because both parties can talk whenever they desire to talk, and one party does not have to wait for the other party to talk before talking. Also, the two parties can talk concurrently.

Embodiments of the claimed invention address problems in the prior art methods where an HTTP transaction includes opening a socket connection, sending an HTTP request, executing an action, formulating HTTP responses, and sending the HTTP responses back to the client. In such prior art methods, the sending of **responses** can only occur in response to client **requests** and, thus, the prior art methods only allow for **synchronous** communication and **not** two-way **asynchronous** communication. (Applicants' FIGs. 2 and 4; Specification; page 3, lines 5-7 and 12-22; page 6, lines 5-24; page 8, line 27 to page 9, line 7).

In column 6, line 55 to column 7, line 17, Rangarajan states the following:

> "The HTTP server 16 is the server side front end which interacts with the clients 24. The HTTP server 16 receives a client **request** and parses it to extract the URL of the requested document.
>
> HTTP servers are designed to serve documents and in most cases do not process data sent from a client, such as data in the form of a cookie. In such a situation, a gateway program is used to process the client data on the server end. In the Internet environment, the Common Gateway Interface ('CGI') is the mechanism which controls the flow of data from the HTTP server to the gateway program. According to the CGI specification, data is sent to the gateway programs through environment variables and read by the program from standard input. To return data back to the HTTP server, the gateway program writes out

the data to its standard output, which is then read by the HTTP server and, after proper modifications to the data headers, returned to the client.

In the present invention, a CGI script 18 is used as an interface between the HTTP server 16 and the State Management Server 12. When a client **request** is received, the HTTP server 16 sets the CGI environment variables to reflect the full URL of the requested document and the cookie(s) accompanying the client's HTTP request. The CGI script 18 is then executed. The script 18 is configured to establish an **Internet socket connection** with the State Management Server 12 and then forward the URL and any received cookies to the SMS 12. The particular implementation of such a CGI script will be apparent to one of skill in the art and is therefore not discussed in detail herein." (Rangarajan; column 6, line 55 to column 7, line 17) (Emphasis Added).

The Examiner points to the **Internet socket connection** established by the CGI script 18 between the CGI script 18 and the State Management Server 12 in the system of Rangarajan as being a **two-way asynchronous** communication connection. (Office Action; page 3). However, the CGI script 18 in the system of Rangarajan does **not** perform operations for two-way **asynchronous** communication with the SMS 12 over the Internet socket connection. (Rangarajan; column 6, line 55 to column 8, line 13).

Instead, in the system of Rangarajan, when the HTTP server 16 receives a client request from the client 24, the following actions occur: (i) the HTTP server 16 parses the request to extract the URL of the requested document; (ii) the HTTP server 16 sets the CGI environment variables to reflect the full URL of the requested document and the cookies accompanying the client's HTTP request; (iii) the CGI script 18 **forwards** the URL and any received cookies to the SMS 12; (iv) the SMS 12 accesses the Registration Table data and determines the file path of the appropriate document for the client to receive according to the data contained in the cookie; (v) the cookie state information is then revised to indicate the new reference and determined file path; (vi) a new cookie is **returned** from the SMS 12 to the CGI script 18; (vii) the new cookie is returned from the CGI script 18 to the HTTP server 16; and (viii) the HTTP server 16 retrieves the identified document and returns it and the modified cookie to the client 24. (Rangarajan; column 6, line 55 to column 7, line 44).

Thus, in the system of Rangarajan, the CGI script 18 **forwards** a URL and any received cookies to the SMS 12 over the Internet socket connection, and then the SMS 12 responds to the data received from the CGI script 18 by **returning** a new cookie or a file path to the CGI script 18 over the Internet socket connection. (Rangarajan; column 7, lines 31-44; column 7, line 66 to column 8, line 13). As a consequence, the CGI script 18 only performs operations for **synchronous** communication over the Internet socket connection with the SMS 12, and does **not** perform operations for two-way **asynchronous** communication with the SMS 12.

Second, Cianfrocca does not cure the deficiency with respect to the teaching of Rangarajan discussed above, because Cianfrocca similarly neither discloses nor suggests a method of maintaining two-way asynchronous communication between a client and a web server using a **single HTTP transaction** in which: (i) an **HTTP request** is communicated from the client to the web server, where the **HTTP request** is configured to initialize a CGI that operates within or in conjunction with the web server; and (ii) operations associated with the CGI are executed, where the operations are configured to perform the two-way asynchronous communication with the client.

The system of Cianfrocca includes a messenger system that is a multi-protocol server that supports HTTP and a **native messenger system protocol**. (Cianfrocca; column 3, line 66 to column 4, line 2). An example of a native messenger system protocol in the system of Cianfrocca is the native Tempest Messenger Protocol (**TMSP**). (Cianfrocca; column 4, lines 2-3). The messenger system in the system of Cianfrocca identifies a protocol that is used for a connection and treats it appropriately. (Cianfrocca; column 4, lines 7-10).

In the case that **HTTP** 1.0 is the protocol used to connect to the messenger system in the system of Cianfrocca, the messenger system **closes** the socket connection once information is sent back across the connection. (Cianfrocca; column 4, lines 11-19). Thus, any socket connection established with a client in response to an **HTTP request** from the client in the system of Cianfrocca is only a **synchronous** socket connection, because the socket connection is **closed** once information is sent back across the connection. Indeed, Cianfrocca states that, "[t]he

nature of a **HTTP request** is that there is request for connection which is made to respond to a single query." (Cianfrocca; column 8, lines 48-50) (Emphasis Added).

In contrast, a method as recited in the present claim allows for maintaining two-way asynchronous communication between a client and a web server using a single **HTTP transaction**. Since the system of Cianfrocca only allows for establishing a **synchronous** socket connection between a messaging system and a client in response to an **HTTP request** from the client, Cianfrocca does not disclose or suggest the features of the present claim.

While a messaging system in the system of Cianfrocca does allow for warm socket connections that are full-duplex and that remain open, such warm socket connections in the system of Cianfrocca can only be established by the messaging system in response to a **Connect function** that is transmitted in a **native messenger system protocol**. (Cianfrocca; column 4, lines 32-39). The warm socket connections in the system of Cianfrocca **cannot** be established between a messaging system and a client in response to an **HTTP** request from the client. Instead, the system of Cianfrocca requires the use of a **native messenger system protocol**, such as the native Tempest Messenger Protocol (**TMSP**). (Cianfrocca; column 4, lines 2-3 and lines 20-47).

In the system of Cianfrocca, a messenger system enabled application uses a supporting library to connect to the messenger system using the **native messenger system protocol**. (Cianfrocca; column 4, lines 20-22). Each connection in the system of Cianfrocca is initiated by the application itself and is terminated by the application itself, and no control over the connection arrangement exists in the messenger system. (Cianfrocca; column 5, lines 34-38). Thus, a warm socket connection in the system of Cianfrocca between a messenger system and a messenger system enabled application can only be established in response to a Connect function transmitted from the messenger system enabled application in a **native messenger system protocol**, such as **TMSP**, and **cannot** be established in response to an **HTTP request** from the messenger system enabled application. (Cianfrocca; column 4, lines 20-47; column 6, lines 50-51).

Thus, even if the teaching of Rangarajan were combined with the teaching of Cianfrocca, the combined teachings would still not disclose or suggest a method of maintaining two-way asynchronous communication between a client and a web server using a **single HTTP transaction**.

Therefore, independent claim 1, as amended, is neither disclosed nor suggested by the Rangarajan and Cianfrocca references and, hence, is believed to be allowable. The Examiner has not made out a *prima facie* case of obviousness under 35 U.S.C. 103.

Independent claim 9, as amended, recites a system for maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction with features similar to features of a method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction of independent claim 1. Therefore, independent claim 9 is believed to be allowable for at least the same reasons that independent claim 1 is believed to be allowable.

Independent claim 20, as amended, recites a method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction with features similar to features of a method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction of independent claim 1. Therefore, independent claim 20 is believed to be allowable for at least the same reasons that independent claim 1 is believed to be allowable.

Independent claim 21, as amended, recites a system for maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction with features similar to features of a method of maintaining two-way asynchronous communication between a client and a web server using a single HTTP transaction of independent claim 1. Therefore, independent claim 21 is believed to be allowable for at least the same reasons that independent claim 1 is believed to be allowable.

The dependent claims are deemed allowable for at least the same reasons indicated above with regard to the independent claims from which they depend. It is noted that, with respect to dependent claim 18, Reisman does not cure the deficiencies with regard to the teachings of Rangarajan and Cianfrocca discussed above.

Conclusion:

Applicants believe that the present application is now in condition for allowance. Favorable reconsideration of the application as amended is respectfully requested.

The Examiner is invited to contact the undersigned by telephone if it is felt that a telephone interview would advance the prosecution of the present application.

The Commissioner is hereby authorized to charge any additional fees which may be required regarding this application under 37 C.F.R. §§ 1.16-1.17, or credit any overpayment, to Deposit Account No. 19-0741. Should no proper payment be enclosed herewith, as by a check being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 19-0741.

If any extensions of time are needed for timely acceptance of papers submitted herewith, Applicants hereby petition for such extension under 37 C.F.R. §1.136 and authorizes payment of any such extensions fees to Deposit Account No. 19-0741.

Respectfully submitted,

Date 2-1-06

By David A. Blumenthal

FOLEY & LARDNER LLP
Customer Number: 23392
Telephone: (202) 672-5407
Facsimile: (202) 672-5399

David A. Blumenthal
Attorney for Applicant
Registration No. 26,257